Optimized Stochastic Policies for Task Allocation in Swarms of Robots

Spring Berman, *Member, IEEE*, Ádám Halász, *Member, IEEE*, M. Ani Hsieh, *Member, IEEE*, and Vijay Kumar, *Fellow, IEEE*

***Abstract*—We present a scalable approach to dynamically allocating a swarm of homogeneous robots to multiple tasks, which are to be performed in parallel, following a desired distribution. We employ a decentralized strategy that requires no communication among robots. It is based on the development of a continuous abstraction of the swarm obtained by modeling population fractions and defining the task allocation problem as the selection of rates of robot ingress and egress to and from each task. These rates are used to determine probabilities that define stochastic control policies for individual robots, which in turn produce the desired collective behavior. We address the problem of computing rates to achieve fast redistribution of the swarm subject to constraint(s) on switching between tasks at equilibrium. We present several formulations of this optimization problem that vary in the precedence constraints between tasks and in their dependence on the initial robot distribution. We use each formulation to optimize the rates for a scenario with four tasks and compare the resulting control policies using a simulation in which 250 robots redistribute themselves among four buildings to survey the perimeters.**

***Index Terms*—distributed control, Markov processes, optimization, stochastic systems, swarm robotics, task allocation**

I. Introduction

ADVANCES in embedded processor, sensor, and actuation technology are paving the way for the development of "swarms" of robots numbered in the hundreds or thousands. We present a strategy for reallocating a swarm of homogeneous robots among a set of tasks that are to be performed in parallel, continuously, and independently of one another. For instance, each task could be an activity at a physical site such as building surveillance, environmental monitoring, construction, or a search-and-rescue operation. The objective is for the robots to autonomously redistribute as quickly and efficiently as possible such that the steady-state populations at the tasks follow a predefined distribution.

This is an instance of the single-task robot, multi-robot task problem (ST-MR) [1], where the goal is to assign teams of robots to tasks in a way that maximizes the system's performance. This is known as the *coalition formation* problem when applied to software agents. Tractable approaches to this problem, which is NP-hard, rely on extensive agent cooperation that is not easily implemented in robot systems since communication can be costly and unreliable and resources are not transferrable [2]. The algorithm in [3] was adapted to the multi-robot domain in [4], but robots must compute all possible coalitions and agree on the best ones, and coalition sizes are limited. The ST-MR problem has recently been addressed with market-based techniques, although allocation strategies for robots have mostly considered the problem of assigning a single robot to each task [2]. Market-based approaches [5] require robots to execute complex bidding schemes based on perceived costs and utilities, and the computation and communication requirements often scale poorly as the number of robots and tasks increases.

These algorithms are not suitable for the large-scale systems that we consider. If tasks are at different sites, communication between all robots may not be possible due to interference, obstruction, or power limitations, or it may be too risky, as in military applications. Also, bandwidth becomes a limiting factor in communication as population size increases. In light of these issues, we propose a strategy that does not use inter-robot communication. We do, however, assume that a central controller broadcasts information about tasks and task transitions without dictating the actions of individual robots.

Our strategy should be readily implementable on robots with limited on-board resources, scalable in the number of robots and tasks, and robust to changes in the population. These properties are inherent in decentralized approaches [6]–[8] that are inspired by the self-organized behavior of social insects such as ants [9]. In this work, robots switch between simple behaviors based on environmental stimuli and interactions with other robots. We adopt this distributed paradigm using stochastic switching between tasks. We note that the potential-based algorithm in [10] is also scalable, but it is designed for tasks that are depleted and does not address the problem of allocating robots as quickly as possible.

Recent work on decentralized control for task allocation has focused on abstracting the physical system to an accurate macroscopic model [11], [12]. Identical robot controllers are defined with stochastic state transitions, and they are averaged to obtain a set of differential or difference equations. System performance is studied by running the model, which is validated through simulations, under many different conditions. We use a controller synthesis approach that is less computationally expensive and gives theoretical guarantees on

Manuscript received July 27, 2008; revised May XX, 2009. This work was supported in part by ARO Grant W911NF-05-1-0219, ONR Grants N00014-08-1-0696 and N00014-07-1-0829, ARL Grant W911NF-08-2-0004, NSF Grant CSR-CPS 0720801, and the NSF Graduate Research Fellowship.

S. Berman and V. Kumar are with the GRASP Laboratory, University of Pennsylvania, 3330 Walnut Street, Philadelphia, PA 19104, USA (e-mail: spring@seas.upenn.edu; kumar@grasp.upenn.edu).

Á. Halász is with the Dept. of Mathematics, West Virginia University, 307G Armstrong Hall, Morgantown, WV 26506, USA (e-mail: halasz@math.wvu.edu).

M. A. Hsieh is with the Dept. of Mechanical Engineering and Mechanics, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104, USA (e-mail: mhsieh1@drexel.edu).

performance. We model the swarm as a set of differential equations in which the variables are continuous population fractions at tasks. We then use standard analysis and optimization tools to design the model parameters so that the swarm macroscopically displays rapid, efficient deployment to the desired distribution. We use these parameters to define rates of switching between tasks that can be realized on individual robots, which collectively display the properties of the continuous model.

We first used this approach to design ant-inspired behaviors that cause robots to converge to the better of two sites [13] or split between two sites in a specified ratio [14]. We extended our methodology to the distribution of a swarm among tasks at many sites [15] and introduced quorum-based stochastic control policies [16]. In the present work, we focus on *optimizing* the rates at which robots switch between tasks for fast convergence to the desired distribution subject to a constraint(s) on idle transitions at equilibrium. We began this investigation in [17], in which we accounted for transition times within the differential equation framework. Here we present several new optimization methods and compare them using a four-site surveillance scenario.

II. Continuous Models

A. Definitions and Assumptions

Consider a population of N robots to be allocated among M tasks. We denote the number of robots performing task $i \in \{1, \ldots, M\}$ at time t by $n_i(t)$, a nonnegative integer, and the desired number of robots for task i by n_i^d, a positive integer. The population fraction performing task i at time t is $x_i(t) = n_i(t)/N$, and the vector of population fractions is $\mathbf{x}(t) = [x_1(t), \ldots, x_M(t)]^T$. The *target distribution* is the set of desired population fractions for each task, $\mathbf{x^d} = [x_1^d \ ... \ x_M^d]^T$, where $x_i^d = n_i^d/N$. A specification in terms of fractions rather than integers is practical for scaling as well as for applications in which losses of robots are common.

The precedence constraints between tasks can be modeled using a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$, the set of M vertices, represents tasks $\{1, \ldots, M\}$ and $\mathcal{E}$, the set of $N_\mathcal{E}$ edges, represents possible transitions between tasks. Tasks i and j are *adjacent*, denoted by $i \sim j$, if a robot that is working on task i can switch to task j. We denote this relation by the ordered pair $(i, j) \in \mathcal{V} \times \mathcal{V}$, with the set $\mathcal{E} = \{(i, j) \in \mathcal{V} \times \mathcal{V} \mid i \sim j \}$. For example, if each task i is an activity at a physical site i, then $\mathcal{G}$ models the site interconnection topology: $\mathcal{V}$ is the set of M sites and each edge (i, j) represents a one-way route that robots can travel from i to j. If there are P possible routes from i to j, then they are represented by distinct edges $(i, j)_m$, $m = 1, ..., P$.

We require $\mathcal{G}$ to be *strongly connected*, which means that a directed path exists between any pair of distinct vertices. This facilitates redistribution by allowing the robots to perform any task starting from any other task; no task acts as a source or a sink. We also consider the case of a *fully connected* graph, in which every vertex is adjacent to every other vertex. This allows robots to switch directly from one task to another rather than working on a sequence of intermediate tasks first.

We consider $\mathbf{x}(t)$ to represent the distribution of the state of a Markov process on $\mathcal{G}$, for which $\mathcal{V}$ is the state space and $\mathcal{E}$ is the set of possible transitions. Every edge (i, j) is assigned a constant positive *transition rate* k_{ij}, the probability per unit time for one robot at task i to switch to task j. These rates define stochastic transition rules: the robots are programmed to switch from task i to j with probability $k_{ij}\delta t$ at each time step δt. The number of transitions between tasks i and j in time Δt has a Poisson distribution with parameter $k_{ij}\Delta t$. Our objective is to compute k_{ij} that cause the robots to quickly redistribute among the tasks in order to occupy them in the population ratios dictated by $\mathbf{x^d}$. The use of *constant* k_{ij} is necessary to abstract the system to a linear continuous model (Sections II-B and II-C), which is used to design the k_{ij} via optimization techniques (Section III).

We assume that a central controller determines $\mathbf{x^d}$, computes the rates k_{ij}, and broadcasts the rates to the robots. The robots have complete knowledge of $\mathcal{G}$ and the tasks to perform; this information can be preprogrammed and updated via a broadcast if the tasks change. The robots must also be capable of executing the tasks and transitions. For instance, if the tasks are at different sites, the robots must be able to localize themselves in their environment and navigate safely between sites.

B. Base Model

The swarm can be modeled as a function of the rates k_{ij} by representing it in terms of the continuous quantity $\mathbf{x}(t)$. In the limit $N \to \infty$, the physical system of individual robots can be abstracted to the following linear ordinary differential equation (ODE) model according to the theoretical justification given in [18],

$$\dot{x}_i(t) = \sum_{\forall j | (j,i) \in \mathcal{E}} k_{ji} x_j(t) - \sum_{\forall j | (i,j) \in \mathcal{E}} k_{ij} x_i(t) \ , \qquad (1)$$

where $i = 1, ..., M$. If there are P edges from i to j, each with rate $k_{ij,m}$, $m \in \{1, ..., P\}$, then $k_{ij} = \sum_{m=1}^{P} k_{ij,m}$.

We define the *flux* from task i to j at time t as $k_{ij}x_i(t)$, the fraction of robots per unit time that are leaving i to switch to j. Hence, model (1) quantifies the rate of change of population fraction $x_i(t)$ as the difference between the total influx and total outflux of robots at task i. The model captures this effect in a simple way by representing robots as switching instantaneously from one task to another, ignoring the time that robots take to effect transitions. Because the k_{ij} are constant, robots still switch between tasks at equilibrium, when the net flux through each task is zero. This contributes to system robustness since the population at each task, which may be depleted by breakdowns, is constantly replenished. The persistent switching may also serve a useful function, such as patrolling or sampling between sites.

Since the number of robots is conserved, system (1) is subject to the constraint

$$\mathbf{1}^T \mathbf{x} = 1 \ . \qquad (2)$$

System (1) can be equivalently written as the linear model

$$\dot{\mathbf{x}} \ = \ -\mathbf{K}\mathbf{x} \ , \qquad (3)$$

where $\mathbf{K} \in \mathbb{R}^{M \times M}$ is a matrix with the properties

$$\mathbf{K}^T \mathbf{1} = \mathbf{0} \ , \tag{4}$$

$$\mathbf{K}_{ij} \leq 0 \quad \forall (i,j) \in \mathcal{E} \ . \tag{5}$$

These two properties result in the following matrix structure:

$$\mathbf{K}_{ij} = \begin{cases} -k_{ji} & \text{if} \quad i \neq j \ , \quad (j,i) \in \mathcal{E} \ , \\ 0 & \text{if} \quad i \neq j \ , \quad (j,i) \notin \mathcal{E} \ , \\ \sum_{(i,l) \in \mathcal{E}} k_{il} & \text{if} \quad i = j \ . \end{cases} \tag{6}$$

Theorem 1: If the graph $\mathcal{G}$ is strongly connected, then system (3) subject to (2) has a unique, stable equilibrium.

Proof: Since $\mathcal{G}$ is strongly connected, the rank of $\mathbf{K}$ is $M-1$ [19]. The null space of $\mathbf{K}$, $\mathbf{x^n}$, is therefore one-dimensional. This null space is intersected by the $(M-1)$-dimensional hyperplane described by constraint (2). Thus, system (3) subject to (2) has a unique equilibrium point, which we call $\bar{\mathbf{x}}^{\mathbf{n}} = [\bar{x}_1^n \ ... \ \bar{x}_M^n]^T$.

Now consider the matrix $\mathbf{T} = t\mathbf{I} - \mathbf{K}$, where $t > 0$ and $\mathbf{I} \in \mathbb{R}^{M \times M}$ is the identity matrix. Choose t large enough such that $\mathbf{T}$ is a nonnegative matrix. Since $\mathcal{G}$ is strongly connected, the matrix $-\mathbf{K}$, and therefore $\mathbf{T}$, is irreducible. Because $\mathbf{T}$ is nonnegative and irreducible, by the Perron-Frobenius theorem $\mathbf{T}$ has a real, positive, simple eigenvalue $\lambda_m(\mathbf{T})$ such that all other eigenvalues of $\mathbf{T}$, $\lambda(\mathbf{T})$, satisfy $|\lambda(\mathbf{T})| < \lambda_m(\mathbf{T})$. This eigenvalue also satisfies the inequalities $\min_j \sum_{i=1}^M \mathbf{T}_{ij} \leq \lambda_m(\mathbf{T}) \leq \max_j \sum_{i=1}^M \mathbf{T}_{ij}$ [19]. Since the columns of $\mathbf{K}$ sum to 0, both sides of these inequalities are t, so $\lambda_m(\mathbf{T}) = t$. Note that $\lambda(\mathbf{T}) = \lambda(-\mathbf{K}) + t$. Thus, the eigenvalue of $-\mathbf{K}$ corresponding to $\lambda_m(\mathbf{T})$ is 0, and all other eigenvalues of $-\mathbf{K}$ satisfy $|\lambda(-\mathbf{K}) + t| < t$. It follows that $-\mathbf{K}$ has a simple zero eigenvalue and all its other eigenvalues satisfy $Re(\lambda(-\mathbf{K})) < 0$. Therefore, the equilibrium point $\bar{\mathbf{x}}^{\mathbf{n}}$ is stable. ■

Theorem 1 proves that system (3) subject to (2) always converges to a single equilibrium $\bar{\mathbf{x}}^{\mathbf{n}}$, which represents the steady-state distribution of population fractions among the M tasks. Hence, we can achieve the target robot distribution $\mathbf{x^d}$ from any initial distribution $\mathbf{x^0}$ by specifying that $\bar{\mathbf{x}}^{\mathbf{n}} = \mathbf{x^d}$ through the following constraint on $\mathbf{K}$:

$$\mathbf{K}\mathbf{x^d} = 0 \ . \tag{7}$$

When the k_{ij} are chosen such that the corresponding $\mathbf{K}$ matrix satisfies constraint (7), a swarm of robots that use the k_{ij} as stochastic transition rules will redistribute from any $\mathbf{x^0}$ to $\mathbf{x^d}$. In practice, this redistribution must take place in a reasonably short amount of time. Since (3) is a linear system, the rate of convergence of $\mathbf{x}$ to $\mathbf{x^d}$ is governed by the real parts of the eigenvalues of $\mathbf{K}$, which are positive homogenous functions of the k_{ij} [20]. Thus, the rate of redistribution can be made arbitrarily fast by using high k_{ij}.

However, in actual robotic systems there is often a substantial cost to using high k_{ij}. At equilibrium, the probability that any robot doing task i will start switching to task j in time step δt is $k_{ij} n_i^d \delta t$. Thus, raising k_{ij} increases the equilibrium "traffic" of robots transitioning between tasks i and j. This switching expends power; for instance, if the tasks are at different locations, the robots must travel between them and may experience delays due to congestion along the route.

Thus, when choosing the k_{ij}, we are faced with a tradeoff between rapid convergence to $\mathbf{x^d}$ and long-term system efficiency, i.e. few idle transitions between tasks once $\mathbf{x^d}$ is achieved. In light of this tradeoff, we frame our objective as the design of an *optimal transition rate matrix* $\mathbf{K}$ that maximizes the convergence rate of system (3) to $\mathbf{x^d}$ subject to one of two possible constraints on task transitions at equilibrium. The first is a limit on the total equilibrium flux of robots switching between tasks:

$$\sum_{(i,j) \in \mathcal{E}} k_{ij} x_i^d \leq c_{tot} \ . \tag{8}$$

This constraint does not dictate how the transitioning population is distributed among edges. An alternative constraint achieves this with a set of limits on the equilibrium flux between each pair of adjacent tasks:

$$k_{ij} x_i^d \leq c_{ij}, \quad (i,j) \in \mathcal{E} \ . \tag{9}$$

C. Time-Delayed Model

As mentioned previously, model (3) does not account for the fact that in reality, the influx of robots to task j from task i is delayed by the time taken to switch between the tasks, τ_{ij}. If we assume a constant transition time τ_{ij} for each edge (i,j), this effect can be included by rewriting equation (1) as a delay differential equation (DDE):

$$\dot{x}_i(t) \ = \sum_{\forall j | (j,i) \in \mathcal{E}} k_{ji} x_j(t - \tau_{ji}) \ - \sum_{\forall j | (i,j) \in \mathcal{E}} k_{ij} x_i(t) \ . \tag{10}$$

where $i = 1, ..., M$. Due to the finite τ_{ij}, there will be robots in the process of switching between tasks; thus, $\sum_{i=1}^M x_i(t) < 1$ for $t > 0$. Let $n_{ij}(t)$ be the number of robots in transition from task i to j at time t and $y_{ij}(t) = n_{ij}(t)/N$. Then the conservation equation for this system is:

$$\sum_{i=1}^M x_i(t) + \sum_{i=1}^M \sum_{\forall j | (i,j) \in \mathcal{E}} y_{ij}(t) \ = \ 1 \ . \tag{11}$$

In practice, robots will complete a transition in different amounts of time, so model (10) can be made more realistic by defining the τ_{ij} as random variables, T_{ij}. In the case where robots effect transitions by traveling between sites, variations in τ_{ij} can arise from changes in navigation patterns caused by collision avoidance, congestion, and localization errors. For this case, we can estimate a reasonable form for the probability density of the T_{ij} from an analogous scenario in which vehicles deliver items along roads to different sites. Vehicle inter-site travel times have been modeled as following an Erlang distribution to capture the fact that the times have positive, minimum possible values; a small probability of being large due to accidents, breakdowns, and low energy; and their distributions tend to be skewed toward larger values [21]. We assume that each T_{ij} follows this distribution with parameters $\omega_{ij} \in \mathbb{Z}^+$ and $\theta_{ij} \in \mathbb{R}^+$:

$$g(t; \omega_{ij}, \theta_{ij}) = \frac{\theta_{ij}^{\omega_{ij}} t^{\omega_{ij}-1}}{(\omega_{ij}-1)!} e^{-\theta_{ij} t} \ . \tag{12}$$

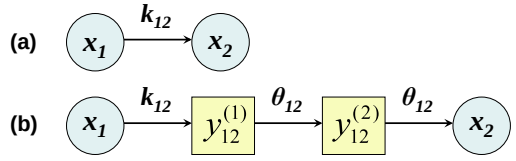

Fig. 1. A labeled edge $(i,j) = (1,2)$ that consists of (a) the real tasks, corresponding to model (1), and (b) both real and virtual tasks (for $\omega_{12} = 2$), corresponding to model (13).

In practice, the parameters are estimated by fitting empirical transition time data to density (12).

Under this assumption, the DDE model (10) can be transformed into an equivalent ODE model of the form (1), which allows us to optimize the rates k_{ij} using the methods we develop for this type of model. We use the fact that T_{ij} has the same distribution as the sum of ω_{ij} independent random variables, $T_1, ..., T_{\omega_{ij}}$, with a common distribution $f(t;\theta_{ij}) = \theta_{ij} e^{-\theta_{ij} t}$ [22]. Each of the variables represents a portion of the transition time between tasks i and j. To model these portions of the transition, we define a directed path composed of a sequence of *virtual tasks*, $u = 1, ..., \omega_{ij}$, between the real tasks i and j. Assume that robots transition instantaneously from virtual task u to $u+1$, which is task j when $u = \omega_{ij}$, at a constant probability per unit time, θ_{ij}. It follows that $f(t;\theta_{ij})$ is the distribution of the time that a robot spends doing virtual task u, and so we can define T_u, $u \in \{1, ..., \omega_{ij}\}$, as this task execution time.

We denote the population fraction that is doing virtual task u along edge (i,j) by $y_{ij}^{(u)}$. Then $\sum_{u=1}^{\omega_{ij}} y_{ij}^{(u)}$ represents y_{ij}, the fraction of robots in transition from task i to task j. Fig. 1 illustrates how an edge from model (1) is expanded with two virtual states $y_{ij}^{(u)}$. As in Section II-B, the dynamics of the population fractions at all real and virtual tasks in the expanded system can be written as a set of linear ODE's:

$$\begin{aligned} \dot{x}_i(t) &= \sum_{j|(j,i)\in\mathcal{E}} \theta_{ji} y_{ji}^{(\omega_{ji})}(t) - \sum_{j|(i,j)\in\mathcal{E}} k_{ij} x_i(t) \ , \\ \dot{y}_{ij}^{(1)}(t) &= k_{ij} x_i(t) - \theta_{ij} y_{ij}^{(1)}(t) \ , \\ \dot{y}_{ij}^{(m)}(t) &= \theta_{ij} \left(y_{ij}^{(m-1)}(t) - y_{ij}^{(m)}(t) \right) \ , \\ & \quad m = 2, ..., \omega_{ij} \ , \end{aligned} \tag{13}$$

where $i = 1, ..., M$ and $(i,j) \in \mathcal{E}$.

We now show that this more realistic model converges to a designable target distribution from any initial distribution.

Theorem 2: If the graph $\mathcal{G}$ is strongly connected, then system (10) subject to (11) with each τ_{ij} distributed according to density (12) has a unique, stable equilibrium.

Proof: We prove this for the equivalent ODE model (13) subject to (11), where $y_{ij} = \sum_{u=1}^{\omega_{ij}} y_{ij}^{(u)}$. Let $\mathbf{y}$ be the vector of $y_{ij}^{(u)}$, $u = 1, ..., \omega_{ij}$, $(i,j) \in \mathcal{E}$. The system state vector is then $\mathbf{z} = [\mathbf{x}\ \mathbf{y}]^T$. We interpret each component of $\mathbf{z}$ as the population fraction at task $i \in \{1, ..., M'\}$, where M' is the sum of all real and virtual tasks. The interconnection topology of these tasks can be modeled as a directed graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, where $\mathcal{V}' = \{1, ..., M'\}$ and $\mathcal{E}' = \{(i,j) \in \mathcal{V}' \times \mathcal{V}' \mid i \sim j\}$. Since $\mathcal{G}$ is strongly connected, so is $\mathcal{G}'$.

Each component of $\mathbf{z}$ evolves according to the ODE

$$\dot{z}_i(t) = \sum_{j|(j,i)\in\mathcal{E}'} \hat{k}_{ji} z_j(t) - \sum_{j|(i,j)\in\mathcal{E}'} \hat{k}_{ij} z_i(t) \ , \tag{14}$$

where each $\hat{k}_{ij}$ is defined by the corresponding coefficient in model (13). The system of equations for all M' tasks can be written in the same form as the linear model (3) using an expanded transition rate matrix $\hat{\mathbf{K}}$, and the conservation constraint (11) can be written as $\mathbf{1}^T \mathbf{z} = 1$. Since the system can be represented in the same form as system (3) subject to (2), Theorem 1 can now be applied to show that there is a unique, stable equilibrium. ■

Remark 1: At equilibrium, the incoming and outgoing flux at each virtual task along the path from task i to j is $k_{ij} x_i$, and so model (13) contains the system $\mathbf{Kx} = \mathbf{0}$. Thus, $\mathbf{x^n}$ in $\mathbf{z^n}$, the null space of $\hat{\mathbf{K}}$, is the same as the null space of $\mathbf{K}$ in the corresponding model (3) that ignores transition times. This shows that the ratio between the equilibrium populations at any two real tasks is the same in both models. However, the equilibrium populations $\bar{x}_i^n \equiv x_i^d$ in model (3) are *higher* than those in model (13) because the conservation constraint for the latter model accounts for robots in transition.

Remark 2: The modeling approach in this section can still be applied when the distribution of T_{ij} is complicated (e.g. multimodal) by approximating it as a combination of Erlang distributions; this is a topic for future work.

III. Design of Optimal Transition Rate Matrix $\mathbf{K}$

We consider the task of redeploying a swarm modeled as system (3) from an initial distribution $\mathbf{x^0}$ to a target distribution $\mathbf{x^d}$. As described in Section II-B, we want to select the rates k_{ij} in a way that balances fast convergence to $\mathbf{x^d}$ with long-term efficiency to conserve power. To this end, we compute the matrix $\mathbf{K}$ as the solution to an optimization problem that maximizes a measure of the convergence rate of system (3) to $\mathbf{x^d}$ subject to constraint (8) or (9) on idle transitions at equilibrium. We quantify the degree of convergence to $\mathbf{x^d}$ by the *fraction of misplaced robots*,

$$\mu(\mathbf{x}) = ||\mathbf{x} - \mathbf{x^d}||_2 \ . \tag{15}$$

We say that one system converges faster than another if it takes less time t_f for $\mu(\mathbf{x})$ to decrease to some small fraction f, such as 0.1, of its initial value $\mu(\mathbf{x^0})$.

We formulate several versions of this optimization problem, summarized in Table I (ROC = rate of convergence), each of which is tailored to an application with a particular combination of properties. The graph $\mathcal{G}$ will be fully connected (*FC*) in addition to strongly connected if there are no physical or logical constraints on the flow of robots between pairs of tasks, such as a path in a disaster area that is only wide enough for robots to travel in one direction. In addition, it may be possible to obtain $\mathbf{x^0}$, for instance by identifying robots in an image from an aerial camera.

Problem P3 is solved using a stochastic optimization method that directly minimizes convergence time. The resulting system is used as a baseline to compare the systems computed by the other problems, which manipulate convergence time by

TABLE I
$\mathbf{K}$ OPTIMIZATION PROBLEMS

Problem	*FC*	$\mathbf{x^0}$	Objective
$\mathbf{P1}, \mathbf{P1_R}$			maximize asymptotic ROC [a]
$\mathbf{P2}$	✓		maximize overall ROC
$\mathbf{P3}$		✓	minimize time to reach $0.1\mu(\mathbf{x^0})$
$\mathbf{P4}$	✓	✓	maximize ROC along $\mathbf{x^d} - \mathbf{x^0}$

[a]Maximizes *all* nonzero eigenvalues of $\mathbf{K}$ when the Markov process on $\mathcal{G}$ is reversible and constraint (9) is used; see Section III-A.

maximizing functions of the eigenvalues of $\mathbf{K}$ using linear or semidefinite programs. Since these types of programs can be solved with methods that have polynomial complexity in the worst case [23], we can efficiently compute the $M \times M$ matrix $\mathbf{K}$ for large M. Thus, our allocation approach scales well with the number of tasks.

Our $\mathbf{K}$ design methods can also be applied to the more realistic model (10) with Erlang-distributed τ_{ij} when it is expressed as an equivalent linear model (13), as in [17].

A. Maximizing the asymptotic rate of convergence

If $\mathcal{G}$ is strongly but not necessarily fully connected and $\mathbf{x^0}$ is unknown, we can designate the asymptotic rate of convergence of system (3) to $\mathbf{x^d}$ as the quantity to maximize. Let $\lambda_i(\mathbf{K})$ signify the eigenvalue of $\mathbf{K}$ with the i^{th} smallest real part of all the eigenvalues. By Theorem 1, $\lambda_1(\mathbf{K}) = 0$ and $\lambda_i(\mathbf{K}) > 0$ for $i = 2, ..., M$. Thus, the asymptotic rate of convergence is governed by $Re(\lambda_2(\mathbf{K}))$. Noting that $\mathbf{K}$ is usually not symmetric, we first find a symmetric matrix $\mathbf{S}$ such that $\lambda_2(\mathbf{S}) \leq Re(\lambda_2(\mathbf{K}))$. We replace the objective function $Re(\lambda_2(\mathbf{K}))$ by $\lambda_2(\mathbf{S})$. We can write this problem as a semidefinite program with a linear matrix inequality that arises from a variational characterization of $\lambda_2(\mathbf{S})$.

Theorem 3: Define $\mathbf{\Pi} = diag(\mathbf{x^d})$, which is invertible since $\mathbf{x^d} > \mathbf{0}$. Let $\mathbf{K}$ be a matrix with the structure in (6). Define the matrices

$$\begin{aligned} \mathbf{N} &= \tfrac{1}{2}(\mathbf{\Pi K}^T + \mathbf{K\Pi}) \ , \qquad (16) \\ \tilde{\mathbf{K}} &= \mathbf{\Pi}^{-1/2}\mathbf{K\Pi}^{1/2} \ , \\ \mathbf{S} &= \tfrac{1}{2}(\tilde{\mathbf{K}} + \tilde{\mathbf{K}}^T) = \mathbf{\Pi}^{-1/2}\mathbf{N\Pi}^{-1/2}. \qquad (17) \end{aligned}$$

Then $\lambda_2(\mathbf{S}) \leq Re(\lambda_2(\mathbf{K}))$.[1]

Denote the vector of all k_{ij} by $\mathbf{k} \in \mathbb{R}^{M^2-M}$, which is the optimization variable. Both constraints on transitions can be written in the form $f(\mathbf{k}) \leq 1$, where $f : \mathbb{R}^{M^2-M} \to \mathbb{R}$ is defined as f_{tot} for constraint (8) and f_{ind} for constraint (9):

$$f_{tot}(\mathbf{k}) = \sum_{(i,j)\in\mathcal{E}} k_{ij}x_i^d, \quad f_{ind}(\mathbf{k}) = \max_{(i,j)\in\mathcal{E}} \{k_{ij}x_i^d/c_{ij}\}. \quad (18)$$

Now we can state the optimization problem as: maximize $\lambda_2(\mathbf{S})$ subject to $f(\mathbf{k}) \leq 1$, $\mathbf{k} \geq \mathbf{0}$. We use an alternate formulation [20]: minimize $f(\mathbf{k})$ subject to $\lambda_2(\mathbf{S}) \geq 1$, $\mathbf{k} \geq \mathbf{0}$. The vector $\mathbf{q} = [(x_1^d)^{1/2} \ ... \ (x_M^d)^{1/2}]^T$ is the eigenvector of $\mathbf{\Pi}^{-1/2}\mathbf{N\Pi}^{-1/2}$ corresponding to the zero eigenvalue. From equation (17) and the characterization of eigenvalues in [24], the constraint $\lambda_2(\mathbf{S}) \geq 1$ can be expressed as:

$$\lambda_2(\mathbf{S}) = \inf_{\substack{||\mathbf{x}||=1 \\ \mathbf{x}^T\mathbf{q}=0}} \mathbf{x}^T\mathbf{\Pi}^{-1/2}\mathbf{N\Pi}^{-1/2}\mathbf{x} \geq \inf_{\substack{||\mathbf{x}||=1 \\ \mathbf{x}^T\mathbf{q}=0}} \mathbf{x}^T(\mathbf{I} - \mathbf{qq}^T)\mathbf{x} \quad (19)$$

The problem can now be posed as **Problem P1**, in which the linear matrix inequality comes from (19).

[P1] minimize $f(\mathbf{k})$
subject to $\mathbf{\Pi}^{-1/2}\mathbf{N\Pi}^{-1/2} \succeq \mathbf{I} - \mathbf{qq}^T, \ \mathbf{k} \geq 0$.

Denote the optimized vector of rates by $\mathbf{k}^*$. If constraint (8) is used, then we can achieve the maximum total flux by multiplying $\mathbf{k}^*$ by $c_{tot}/f_{tot}(\mathbf{k}^*)$. If constraint (9) is used, we can achieve the maximum flux for each edge by dividing $\mathbf{k}^*$ by $f_{ind}(\mathbf{k}^*)$.

For a strongly but not necessarily fully connected graph with bidirectional edges, which means that $(i,j) \in \mathcal{E}$ if and only if $(j,i) \in \mathcal{E}$, we explore the advantage of having a *reversible* Markov process, which is defined by the *detailed balance equations*:

$$k_{ij}x_i^d = k_{ji}x_j^d \quad \forall (i,j) \in \mathcal{E} \ . \qquad (20)$$

Suppose that $\mathcal{G}$ has bidirectional edges and the two edges between each pair of adjacent tasks have equal flux capacities. For example, robots may travel between sites along identical parallel roads, similar to a two-way highway. Then by condition (20), the Markov process on $\mathcal{G}$ is reversible. We adapt the problem of maximizing the asymptotic rate of convergence to this special case and call it **Problem $\mathbf{P1_R}$**.

For constraint (8): Condition (20) implies that $\mathbf{K\Pi} = \mathbf{\Pi K}^T$, so $\mathbf{N} = \mathbf{K\Pi}$ in equation (16). Substitute $\mathbf{K\Pi}$ for $\mathbf{N}$ in Problem P1 (with $f = f_{tot}$). Since $\mathbf{K} = \mathbf{N\Pi}^{-1}$, $\mathbf{K}$ is similar to $\mathbf{S}$, so the constraint $\lambda_2(\mathbf{S}) \geq 1$ becomes $\lambda_2(\mathbf{K}) \geq 1$. Thus, the problem constrains $Re(\lambda_2(\mathbf{K}))$ directly instead of a lower bound on this value.

For constraint (9): We can maximize *all* the nonzero eigenvalues of $\mathbf{K}$ by setting each transition rate to its maximum value subject to condition (20) and constraint (9):

$$k_{ij} = (1/x_i^d)\min(c_{ij}, c_{ji}) \ , \quad (i,j) \in \mathcal{E} \ .$$

This is evident by using the Courant-Fischer min-max theorem [24] to express each nonzero eigenvalue of $\mathbf{S}$, and therefore of $\mathbf{K}$, in terms of a quadratic form $\mathbf{x}^*\mathbf{Sx}$ ($\mathbf{x}^*$ is the conjugate transpose of $\mathbf{x}$), which is equal to

$$\sum_{(i,j)\in\mathcal{E}} k_{ij}x_i^d a_{ij}\bar{a}_{ij} \ , \quad a_{ij} = x_i(x_i^d)^{-1/2} - x_j(x_j^d)^{-1/2} \ ,$$

where $\bar{a}_{ij}$ is the complex conjugate of a_{ij}.

B. Maximizing the overall convergence rate

The asymptotic rate of convergence only dictates the long-term system behavior. If $\mathcal{G}$ is fully connected and $\mathbf{x^0}$ is unknown, we can speed convergence of the faster modes by maximizing a measure of the overall convergence rate,

[1]Proofs for the theorems in this section can be found in Appendix A.

which is a function of *all* the nonzero eigenvalues of $\mathbf{K}$, $\mathbf{\Lambda}(\mathbf{K}) = [\lambda_2(\mathbf{K}) \ ... \ \lambda_M(\mathbf{K})]$. We define the quantity to be maximized as $\mathbf{1}^T\mathbf{\Lambda}$, which weights each eigenvalue equally. We use equations (4) and (7) to write $\mathbf{k}$ as a linear function of $\mathbf{v} \equiv [\mathbf{\Lambda}(\mathbf{K}) \ \mathbf{0}]^T \in \mathbb{R}^{M^2-M}$. This allows us to formulate the optimization problem as a linear program with optimization variable $\mathbf{v}$ and objective function $\mathbf{1}^T\mathbf{v}$.

Let $\mathbf{K}$ be a matrix that satisfies equation (4), which sets M constraints on the M^2 entries of $\mathbf{K}$, and equation (7), which sets $M-1$ constraints. We now define the remaining $(M-1)^2$ constraints on $\mathbf{K}$ in terms of the variable $\mathbf{\Lambda}(\mathbf{K})$. Since no extra constraints can be applied, no k_{ij} may be set to zero, which is why $\mathcal{G}$ must be fully connected.

Construct an orthonormal basis set in $\mathbb{R}^M$, $D = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_{M-1}, \mathbf{x^d}/||\mathbf{x^d}||\}$. Define a matrix in $\mathbb{R}^{M\times M}$,

$$\mathbf{A} \ = \ [\ \mathbf{d}_1 \ ... \ \mathbf{d}_{M-1} \ \mathbf{1}\ ]^T \ \equiv \ [\ \tilde{\mathbf{A}}^T \mid \mathbf{1}\ ] \ . \qquad (21)$$

Since $\mathbf{1}^T\mathbf{x^d} = 1$ by equation (2), $\mathbf{1}$ has a nonzero component in the direction of $\mathbf{x^d}$, so the rows of $\mathbf{A}$ are linearly independent. Thus, $\mathbf{A}$ is invertible. Let $\mathbf{B} = \mathbf{A}^{-1}$. Then

$$\mathbf{B} = \left[\ \tilde{\mathbf{A}}^T \mid \mathbf{x^d}\ \right] \left[\ \begin{array}{cc} \mathbf{I} & \mathbf{0} \\ -\mathbf{1}^T\tilde{\mathbf{A}}^T & 1 \end{array}\ \right] \equiv \left[\ \tilde{\mathbf{B}} \mid \mathbf{x^d}\ \right] \ .$$

Define $\mathbf{C} \in \mathbb{R}^{(M-1)\times(M-1)}$ as follows for some fixed $\tilde{\mathbf{A}}$:

$$\mathbf{C} = \tilde{\mathbf{A}}\mathbf{K}\tilde{\mathbf{B}} \ . \qquad (22)$$

Also define $\hat{\mathbf{C}} \in \mathbb{R}^{M\times M}$ as $\mathbf{C}$ augmented with an added row of zeros and an added column of zeros.

Theorem 4: A matrix $\mathbf{K}$ can be expressed as $\mathbf{K} = \mathbf{B}\hat{\mathbf{C}}\mathbf{A}$ if and only if it satisfies equations (4) and (7).

From this result, $\mathbf{K}$ is similar to $\hat{\mathbf{C}}$, and so the eigenvalues of $\mathbf{C}$ are $\mathbf{\Lambda}(\mathbf{K})$. Thus, we can define $\mathbf{C}$ as:

$$\mathbf{C} \equiv diag(\mathbf{\Lambda}(\mathbf{K})) \ . \qquad (23)$$

Now reformulate equation (7) as $\mathbf{Fk} = \mathbf{0}$, where $\mathbf{F} \in \mathbb{R}^{M\times(M^2-M)}$, and equation (22) with $\mathbf{C}$ determined by (23) as $\mathbf{Gk} = \mathbf{g}$, where $\mathbf{G} \in \mathbb{R}^{(M-1)^2\times(M^2-M)}$ and $\mathbf{g} = [\mathbf{\Lambda}(\mathbf{K}) \ \mathbf{0}]^T \in \mathbb{R}^{(M-1)^2}$. Define $\tilde{\mathbf{F}}$ as any $M-1$ rows of $\mathbf{F}$. Then $\mathbf{k}$ can be written as

$$\mathbf{k} = [\mathbf{G} \ \ \tilde{\mathbf{F}}]^{-T}[\mathbf{g}^T \ \mathbf{0}]^T \equiv \mathbf{H}^{-1}\mathbf{v} \ . \qquad (24)$$

Using definition (24) for $\mathbf{k}$, constraints (8) and (9) are

$$\mathbf{r}^T\mathbf{H}^{-1}\mathbf{v} \le c_{tot} \ , \qquad \mathbf{H}^{-1}\mathbf{v} \le \mathbf{c} \ , \qquad (25)$$

where the entries of $\mathbf{r} \in \mathbb{R}^{M^2-M}$ are x_i^d and the entries of $\mathbf{c} \in \mathbb{R}^{M^2-M}$ are c_{ij}/x_i^d. In addition, property (5) is

$$\mathbf{H}^{-1}\mathbf{v} \ge \mathbf{0} \ . \qquad (26)$$

Note that while this property is not needed to prove Theorem 4, it is required to produce a valid $\mathbf{K}$. The optimization problem can now be posed as **Problem P2**.

[P2] Maximize $\mathbf{1}^T\mathbf{v}$ subject to $\mathbf{v}_i = 0$ for $i = M, ..., M^2 - M$, equation (26), and one of the constraints in (25).

C. Maximizing the convergence rate for a specified $\mathbf{x^0}$

If $\mathcal{G}$ is strongly but not necessarily fully connected and $\mathbf{x^0}$ is known, we can use a stochastic optimization method to *directly* minimize the time to converge from $\mathbf{x^0}$, quantified by t_f. We implement **Problem 3** below using Metropolis optimization [25] with $\mathbf{k}$ as the variable. We chose this method for its simplicity and the fact that it yields reasonable improvements in t_f with moderate computing resources.

[P3] Minimize t_f subject to equations (4), (5), (7), and constraint (8) or (9).

Implementation: At each iteration, $\mathbf{k}$ is perturbed by a random vector such that the resulting $\mathbf{K}$ matrix satisfies (4), (5), and (7). $\mathbf{k}$ is then scaled as in Problem P1 to satisfy constraint (8) or (9) while maximizing flux capacity. The resulting $\mathbf{K}$ is decomposed into its normalized eigenvectors and eigenvalues, system (3) is mapped into the space spanned by the normalized eigenvectors, and the appropriate transformation is applied to compute $\mathbf{x}(t)$ using $\exp(t \ diag([\mathbf{\Lambda}(\mathbf{K}) \ 0]))$. Since the system is stable by Theorem 1, $\mu(\mathbf{x})$ always decreases monotonically with time, so a Newton scheme can be used to calculate t_f.

If $\mathcal{G}$ is fully connected and $\mathbf{x^0}$ is known, then $\mathbf{K}$ can be computed such that $\Delta \equiv \mathbf{x^d} - \mathbf{x^0}$ is one of its eigenvectors with eigenvalue $\lambda > 0$. By maximizing λ, we maximize the convergence rate along the vector from $\mathbf{x^0}$ to $\mathbf{x^d}$, the most direct route in $\mathbb{R}^M$ to the target distribution. We use the decomposition of $\mathbf{K}$ from Theorem 4 to formulate the optimization problem as a linear program that maximizes λ.

Theorem 5: Let $\mathbf{K}$ be a matrix that satisfies equations (4) and (7); then by Theorem 4, $\mathbf{K} = \mathbf{B}\hat{\mathbf{C}}\mathbf{A}$. Let $\mathbf{d}_1 = \mathbf{d}$ in definition (21), where

$$\mathbf{d} = \Delta'/||\Delta'||, \quad \Delta' = \Delta - \left(\mathbf{x^d}^T\Delta/||\mathbf{x^d}||^2\right)\mathbf{x^d} \ . \qquad (27)$$

Then $\mathbf{K}\Delta = \lambda\Delta$ if and only if $\mathbf{C}$ from (22) is defined as

$$\mathbf{C} = [\ \mathbf{c} \mid \tilde{\mathbf{C}}\ ], \quad \mathbf{c}^T = [\lambda \ \mathbf{0}] \ , \qquad (28)$$

where λ and $\tilde{\mathbf{C}}$ are unconstrained.

We can now pose the optimization problem as **Problem P4**, in which property (5) and constraints (8) and (9) are defined in terms of the entries of $\mathbf{B}\hat{\mathbf{C}}\mathbf{A}$, with $\mathbf{d_1} = \mathbf{d}$ and $\mathbf{C}$ defined by (28). The optimization variables are λ and $\tilde{\mathbf{C}}$.

[P4] Maximize λ subject to equation (5) and constraint (8) or (9).

IV. Results

A. Effect of connectivity of $\mathcal{G}$ on asymptotic convergence rate

As a preliminary study, we investigated the effect of the connectivity of $\mathcal{G}$ on $\lambda_2(\mathbf{K})$ for several strongly connected, directed graphs on three tasks, labeled in Fig. 2. We used Problem $\mathrm{P1_R}$ to compute $\mathbf{K}$ for graph α with reversibility condition (20) and Problem P1 to compute $\mathbf{K}$ for graph α without this condition and for all other graphs. We modeled each edge in a graph as providing one unit of equilibrium flux capacity by defining $c_{ij} = 1$ for all $(i,j) \in \mathcal{E}$ in constraint (9)

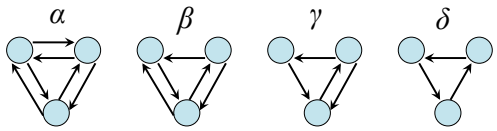

Fig. 2. Graphs on three tasks

TABLE II
COMPARISON OF $\lambda_2(\mathbf{K})$ FOR GRAPHS ON THREE TASKS

Graph	Rev.	$\lambda_2(\mathbf{K})$, constraint (8)	$\lambda_2(\mathbf{K})$, constraint (9)
α	yes	9.6774	7.7299
α	no	$9.6774 \pm 0.0026i$	7.7299
β	no	$8.0645 \pm 2.7936i$	$4.9588 \pm 1.6378i$
γ	no	$6.5729 \pm 2.9691i$	$4.6667 \pm 2.2111i$
δ	no	$5.1667 \pm 2.5766i$	$5.1667 \pm 2.5766i$

and $c_{tot} = N_{\mathcal{E}}$ in constraint (8). The target distribution was $x_1^d = 0.2$, $x_2^d = 0.3$, $x_3^d = 0.5$.

Table II gives the resulting $\lambda_2(\mathbf{K})$ of each graph for both constraints, with column 2 indicating whether condition (20) was imposed. The fully connected graph α yields the fastest convergence, which is expected since robots can switch from any task directly to any other task. Each removal of an edge from graph α lowers $\lambda_2(\mathbf{K})$, except in the case of constraint (9) applied to the 3-edge cycle δ. This is because the optimization problem maximized the flux capacity for *each* edge of graph δ (and did not for β and γ), which offset the stricter limits on task switching than in the other graphs.

B. Comparison of $\mathbf{K}$ for surveillance simulation

To demonstrate our approach on a realistic application, we simulated a scenario in which each task is the surveillance of one of four buildings on the University of Pennsylvania campus. Robots execute the tasks by monitoring the building perimeters, and they effect task transitions by traveling between buildings. We assume that robots can localize themselves on the campus and sense neighboring robots to avoid collisions. Appendix B describes our simulation methodology in detail, including the motion controllers for perimeter tracking and navigation that are used to implement surveillance behavior and inter-site travel, respectively. Fig. 3 illustrates the integration of switching initiations, perimeter surveillance, and navigation in the simulation.

The buildings to be monitored are highlighted in light dashed lines in Fig. 4, which also shows the cell decomposition used for navigation (see Appendix B). Two different graphs $\mathcal{G}$, shown in Fig. 5, were defined on these buildings. The swarm consists of 250 homogeneous robots and is initially split equally between buildings 3 and 4. The robots must redistribute to achieve the target population fractions $x_1^d = 0.1$, $x_2^d = 0.4$, $x_3^d = 0.2$, $x_4^d = 0.3$.

We compared the system convergence to $\mathbf{x^d}$ for different sets of transition rates $\mathbf{k}$, each computed from one of the optimization problems discussed in Section III. Problems P1 and P3 were used to compute rates for the system with graph Fig. 5a, and Problems P1_R, P2, P3, and P4 were used for the system with graph Fig. 5b. The snapshots in Fig. 6 illustrate the robot redistribution for one trial.

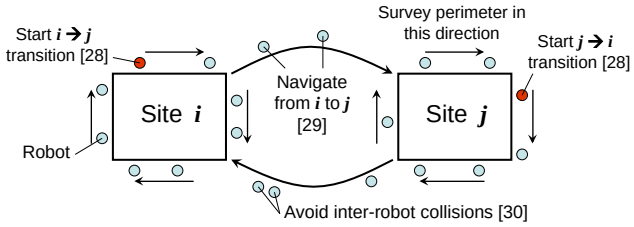

Fig. 3. Robot activities in the simulation. Numbers in brackets are related references for the stochastic simulation algorithm and motion controllers (see Appendix B).

The following discussion summarizes several key points from the simulation results.

a) Agreement between continuous and stochastic models: Our top-down methodology relies on the principle that the rates k_{ij} designed for the continuous model (3) will produce similar system performance when used as stochastic transition probabilities by individual robots. In Fig. 7, we compare performance in terms of $||\mathbf{x}-\mathbf{x^d}||_1$ for 40 stochastic simulation runs and the continuous DDE model (10) with the same $\mathbf{k}$. Each time delay τ_{ij} in the DDE model was estimated as the average of $750-850$ robot travel times at equilibrium from site i to j, collected from a stochastic simulation using site graph Fig. 5b. The stochastic runs average to a plot that is close to the DDE plot and display little variability; if the number of robots were to approach infinity, the standard deviations would decrease to zero. The similarity in performance of the continuous and stochastic models verifies the validity of our top-down methodology.

b) Tradeoff between convergence rate and equilibrium traffic: Fig. 8 and 9 compare system performance for different $\mathbf{k}$ in terms of the *distance from equilibrium*,

$$\nu(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{x^d}||_1 - \mathbf{1}^T\mathbf{y} \ . \tag{29}$$

This quantity decreases to zero at equilibrium because then the fraction of travelers, $\mathbf{1}^T\mathbf{y}$, accounts entirely for all the discrepancies $|x_i - x_i^d|$, $i = 1, ..., M$. Each plot is an average over 40 stochastic simulation runs, and the bold numbers beside the legends are the average traveler fractions at equilibrium for each $\mathbf{k}$. (Standard deviations are not shown to avoid cluttering the figures; the maximum standard deviation over all plots is 0.078.) The data in these figures verify that there is a tradeoff between rapid convergence to equilibrium and the number of idle transitions between sites at equilibrium. For instance, the runs in Fig. 8b are the slowest to converge, and they yield the lowest equilibrium traffic fractions. It is notable that this tradeoff can occur to different degrees depending on the flux constraint, (8) or (9). The P2 plot converges slightly faster in Fig. 9b than in Fig. 9a, but it has a *lower* equilibrium traffic fraction.

c) Faster convergence with increased site connectivity: Fig. 8 and 9 show that for both flux constraints, the runs for graph Fig. 5b converge faster to equilibrium than those

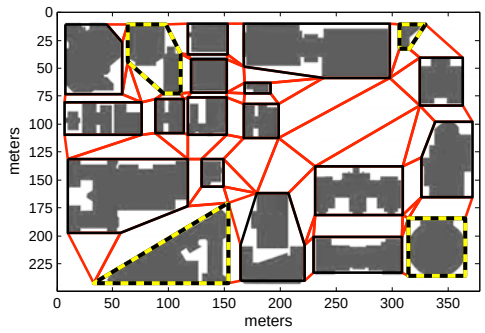

Fig. 4. Cell decomposition of the free space used for navigation.

for graph Fig. 5a. This is due to the difference in allowable pathways between the initial and final distributions. In Fig. 5b, robots can travel directly from sites 3 and 4 to sites 1 and 2, while in Fig 5a, they can only reach sites 1 and 2 via the path $3 \rightarrow 4 \rightarrow 1 \rightarrow 2$, which prolongs the redistribution process. The greater number of edges in Fig. 5b also reduces the impact on convergence of limiting each edge's flux capacity. The range of convergence times to equilibrium for Fig. 5b are similar for both constraints, while the convergence times for Fig. 5a increase significantly when constraint (9) is applied.

d) Limits on edge flux capacities eliminate the advantage of knowing $\mathbf{x^0}$*:* Since the $\mathbf{k}$ produced by Problems P3 and P4 are optimized for a specific $\mathbf{x^0}$, it seems likely that for any given $\mathbf{x^0}$ the P3 and P4 runs will converge at least as fast as the runs corresponding to other problems, which optimize $\mathbf{k}$ for the entire domain of $\mathbf{x^0}$. As Fig. 8a and 9a show, this is true if constraint (8) is used. This is because the flux capacity can be distributed among edges in any way as long as total capacity does not exceed a limit. However, when constraint (9) is used, limits are placed on edges that, if left unconstrained, would be allocated a higher flux capacity to redistribute robots from $\mathbf{x^0}$ to $\mathbf{x^d}$. The problems that are independent of $\mathbf{x^0}$ are more robust to these limitations; their corresponding runs converge as fast as the runs that rely on $\mathbf{x^0}$ or outperform them.

e) $\mathbf{K}$ *from convex optimization is competitive compared to* $\mathbf{K}$ *from stochastic optimization:* The fastest-converging runs that use $\mathbf{k}$ from Problems P1, P1_R, P2, and P4 attain equilibrium at least as quickly as the corresponding runs that use $\mathbf{k}$ from Problem P3. Hence, we can use efficient convex optimization techniques to compute a $\mathbf{k}$ that yields the same system performance as a $\mathbf{k}$ from a much more time-consuming stochastic optimization approach.[2] This facilitates the quick computation of $\mathbf{k}$ in real-time task allocation scenarios.

V. Conclusions

We have presented an approach to redistributing a swarm of robots among multiple tasks that can be optimized for fast convergence to a target distribution subject to a constraint(s) on idle transitions at equilibrium. Our methodology is based on modeling the swarm as a set of continuous linear ODE's and optimizing the transition rates in this model. We can account for realistic distributions of transition times within the framework of the linear ODE abstraction by augmenting the network of tasks with virtual tasks that represent the progress of transitioning robots. The optimized rates comprise a list of $N_\mathcal{E}$ transition probabilities per unit time for individual robots to switch between tasks, and they are independent of the swarm size. The collective behavior that arises from individual robots switching stochastically between tasks follows the continuous model prediction. In this way, we synthesize decentralized robot controllers that can be computed a priori, do not require inter-robot communication, and have provable

[2]On a standard 2 GHz laptop, one Metropolis optimization run used for graph Fig. 5b took about 15 minutes for $t_{0.1}$ to decrease slowly enough with each iteration for $\mathbf{K}$ to be deemed close enough to optimal, while all the convex optimization programs computed an optimal $\mathbf{K}$ in less than a second.

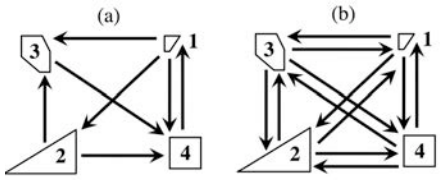

Fig. 5. Numbering and connectivity of surveyed buildings for (a) a strongly connected but not fully connected graph; (b) a fully connected graph.

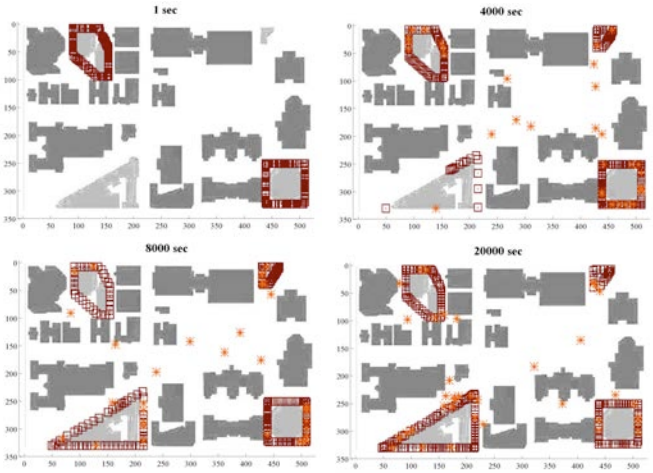

Fig. 6. Snapshots of a run using $\mathbf{k}$ from Problem P1 with constraint (9). The red robots (□) are not engaged in a transition; the orange robots (∗) have committed to travel to another site or are in the process of traveling.

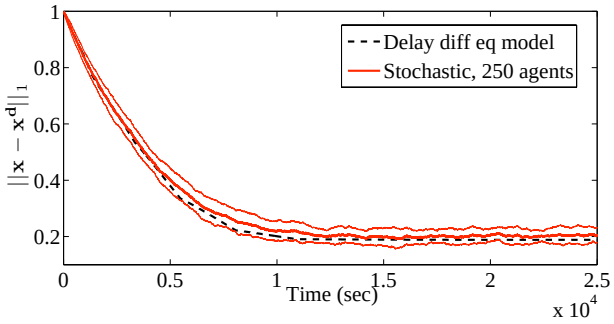

Fig. 7. DDE and stochastic simulations using $\mathbf{k}$ from Problem P1 with constraint (9). Stochastic plots show the average over 40 runs ± standard deviation.

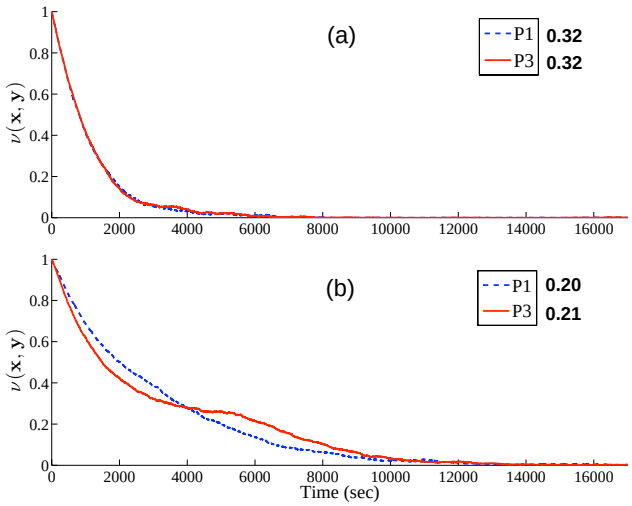

Fig. 8. Distance from equilibrium for stochastic simulations using graph Fig. 5a with (a) constraint (8) and (b) constraint (9). Each plot is an average over 40 runs that use $\mathbf{k}$ from the problem labeled in the legend. The bold number to the right of each legend entry is the equilibrium traveler fraction averaged over 1000 data points of the corresponding plot.

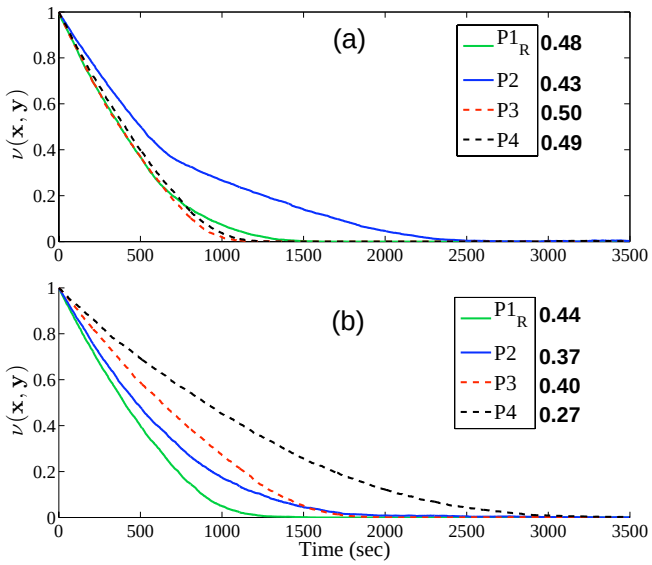

Fig. 9. The same quantities as in Fig. 8 for runs using graph Fig. 5b.

guarantees on performance. A possible extension of this work is the design of a time-dependent transition rate matrix $\mathbf{K}(t)$ that causes the swarm to redistribute according to a trajectory of desired configurations, $\mathbf{x}^{\mathbf{d}}(t)$. Other extensions include the introduction of communication between robots and the use of nonlinear models to represent robot interactions.

APPENDIX A
PROOFS FOR SECTION III

A. Proof of Theorem 3

Define a convex, symmetric function $h : \mathbb{R}^M \to \mathbb{R}$,

$$h(\mathbf{x}) = -\min\{x_i + x_j\} \ , \quad i,j \in \{1,...,M\} \ . \tag{30}$$

Let $\lambda(\mathbf{A})$ be the vector of the eigenvalues of a matrix $\mathbf{A}$. By Theorem 16.4 of [26], since h is convex and symmetric, $h(Re(\lambda(\mathbf{K})))$ is the infimum of $h(\frac{1}{2}\lambda(\mathbf{M}+\mathbf{M}^T))$ over all matrices $\mathbf{M}$ similar to $\mathbf{K}$. Thus, since $\tilde{\mathbf{K}}$ is similar to $\mathbf{K}$,

$$h(Re(\lambda(\mathbf{K}))) \leq h(\tfrac{1}{2}\lambda(\tilde{\mathbf{K}}+\tilde{\mathbf{K}}^T)) = h(\lambda(\mathbf{S})) \ , \tag{31}$$

where the equality on the right comes from equation (17).

Now we evaluate both sides of inequality (31). By Theorem 1, $h(Re(\lambda(\mathbf{K}))) = -Re(\lambda_2(\mathbf{K}))$. We observe that $\lambda(\mathbf{S}) = Re(\lambda(\mathbf{S}))$ because $\mathbf{S}$ is symmetric. We now show that $\mathbf{S}$ is positive semidefinite, denoted by $\mathbf{S} \succeq 0$, which implies that $h(\lambda(\mathbf{S})) = -\lambda_2(\mathbf{S})$ and hence reduces (31) to the inequality $\lambda_2(\mathbf{S}) \leq Re(\lambda_2(\mathbf{K}))$. By equation (17), $\mathbf{S} \succeq 0$ if $\mathbf{N} \succeq 0$. Since $\mathcal{G}$ is strongly connected, $\lambda_2(\mathbf{N}) > 0$ (Lemma 10 of [27]). Using property (4) and constraint (7), $\mathbf{N1} = \frac{1}{2}(\mathbf{\Pi K}^T\mathbf{1} + \mathbf{K x^d}) = \mathbf{0}$, and so $\lambda_1(\mathbf{N}) = 0$ with corresponding eigenvector $\mathbf{1}$. Therefore, $\mathbf{N} \succeq 0$.

B. Proof of Theorem 4

$\mathbf{K}$ is similar to $\mathbf{P} \equiv \mathbf{MKN}$, where $\mathbf{M}, \mathbf{P} \in \mathbb{R}^{M\times M}$ and $\mathbf{N} = \mathbf{M}^{-1}$. Subdivide $\mathbf{M}$ as $[\tilde{\mathbf{M}}^T \mid \mathbf{m}]^T$ and $\mathbf{N}$ as $[\tilde{\mathbf{N}} \mid \mathbf{n}]$, where $\mathbf{m}, \mathbf{n} \in \mathbb{R}^{M\times 1}$. Then

$$\mathbf{MN} = \begin{bmatrix} \tilde{\mathbf{M}}\tilde{\mathbf{N}} & \tilde{\mathbf{M}}\mathbf{n} \\ \mathbf{m}^T\tilde{\mathbf{N}} & \mathbf{m}^T\mathbf{n} \end{bmatrix} = \mathbf{I} \ , \tag{32}$$

$$\mathbf{MKN} = \begin{bmatrix} \tilde{\mathbf{M}}\mathbf{K}\tilde{\mathbf{N}} & \tilde{\mathbf{M}}\mathbf{K}\mathbf{n} \\ \mathbf{m}^T\mathbf{K}\tilde{\mathbf{N}} & \mathbf{m}^T\mathbf{K}\mathbf{n} \end{bmatrix} = \mathbf{P} \ . \tag{33}$$

Choose an $\mathbf{N}$ with $\mathbf{n} = \mathbf{x^d}$. It follows from equation (32) that $\mathbf{m}^T\mathbf{x^d} = 1$, which by equation (2) implies that $\mathbf{m} = \mathbf{1}$.

Suppose that $\mathbf{K}$ satisfies equations (4) and (7). Since $\mathbf{m} = \mathbf{1}$ and $\mathbf{n} = \mathbf{x^d}$, these constraints applied to equation (33) make the last row and last column of $\mathbf{P}$ both $\mathbf{0}$. To satisfy $\tilde{\mathbf{M}}\mathbf{n} = \tilde{\mathbf{M}}\mathbf{x^d} = \mathbf{0}$ in equation (32), $\tilde{\mathbf{M}}$ can be set to $\tilde{\mathbf{A}}$. Then $\mathbf{M} = \mathbf{A}$, $\mathbf{N} = \mathbf{B}$, and $\mathbf{P} = \hat{\mathbf{C}}$, so it follows that $\mathbf{K} = \mathbf{B}\hat{\mathbf{C}}\mathbf{A}$.

Now suppose that $\mathbf{K} = \mathbf{B}\hat{\mathbf{C}}\mathbf{A}$. Since $\hat{\mathbf{C}}\mathbf{A}\mathbf{x^d} = \mathbf{0}$ and $\mathbf{1}^T\mathbf{B}\hat{\mathbf{C}} = \mathbf{0}$, $\mathbf{K}$ satisfies equations (4) and (7).

C. Proof of Theorem 5

Suppose that $\mathbf{K}\Delta = \lambda\Delta$. Then

$$\mathbf{K}\Delta = \mathbf{B}\hat{\mathbf{C}}\mathbf{A}\Delta = \lambda\Delta \quad \Rightarrow \quad \hat{\mathbf{C}}\mathbf{A}\Delta = \lambda\mathbf{A}\Delta \ . \tag{34}$$

Using equation (27) for $\mathbf{d}_1$ and the orthonormality of the $\mathbf{d}_i$:

$$\mathbf{d}_i^T\Delta = ||\Delta'||\mathbf{d}_i^T\mathbf{d}_1 + \left(\mathbf{x^d}^T\Delta/||\mathbf{x^d}||^2\right)\mathbf{d}_i^T\mathbf{x^d} = 0 \tag{35}$$

for $i = 2,...,M-1$. From this equation and the fact that $\mathbf{1}^T\Delta = 0$ by constraint (2), $\mathbf{A}\Delta = [\ \mathbf{d}_1^T\Delta \mid \mathbf{0}\ ]^T$. Thus, equation (34) is true if and only if $\mathbf{C}$ is defined as in (28).

APPENDIX B
SIMULATION METHODOLOGY

The continuous DDE model (10) was numerically integrated using the Runge-Kutta method. Gillespie's Direct Method [28] was used to perform a stochastic simulation of the system that is represented deterministically by the DDE model. This method simulates a sequence of robot transition events and their initiation times using the transition rates k_{ij}. Each event is identified with the commitment of a robot to travel to

another site. A transition from building i to building j is assigned to a random robot on the perimeter of i. This robot continues to track the perimeter until it reaches the exit for edge (i, j), at which point it begins navigating to the entrance on j. For more details, see [13] and [16].

To simulate perimeter tracking and navigation, we represented each robot k as a planar agent governed by a kinematic model $\dot{\mathbf{q}}_k = \mathbf{u}_k$, where $\mathbf{q}_k \in \mathbb{R}^2$ denotes the robot's (x, y) coordinates and $\mathbf{u}_k \in \mathbb{R}^2$ is a control input.

Suppose that the boundary of a building m is parameterized by a vector $\mathbf{s}(s) \in \mathbb{R}^2$ that maps arc length s to (x, y) coordinates. A robot k monitoring the perimeter of m moves in the direction of a unit vector tangent to this boundary, $\hat{\mathbf{n}}_m(\mathbf{s}) \in \mathbb{R}^2$. To create an approximately uniform distribution of robots around the perimeter, we specify that the robot k slows down by a fraction ζ of its nominal speed v_p if its distance q_{kl} from the robot l in front of it is less than p_m/n_m, where p_m is the perimeter length and n_m is the site population. The robot kinematics are then defined as

$$\dot{\mathbf{q}}_k = (1 - \sigma(q_{kl}, p_m, n_m)\zeta)\ v_p\ \hat{\mathbf{n}}_m(\mathbf{q}_k)\ ,$$

where $\sigma(q_{kl}, p_m, n_m) = 1$ if $q_{kl} < p_m/n_m$ and 0 otherwise.

To implement inter-site navigation, we first decomposed the free space into a tessellation of convex cells. Each edge (i, j) was defined as a sequence of cells to be traversed by robots moving from an exit point on the perimeter of building i to an entry point on the perimeter of j. Dijkstra's algorithm was used a priori to compute the sequence with the shortest cumulative distance between cell centroids, starting from the cell adjacent to the exit at i and ending at the cell adjacent to the entrance at j. The robots are provided with the cell sequence corresponding to each edge.

Define $\mathcal{N}_k$ as the set of robots within the sensing radius ρ of robot k. The robot kinematics for navigation are

$$\dot{\mathbf{q}}_k = v_n\ (\mathbf{n}_g(\mathbf{q}_k) + \mathbf{n}_a(\mathbf{q}_k, \mathcal{N}_k))/||\mathbf{n}_g(\mathbf{q}_k) + \mathbf{n}_a(\mathbf{q}_k, \mathcal{N}_k)||\ ,$$

where vector $\mathbf{n}_g(\mathbf{q}_k)$ is computed from local potential functions to ensure arrival at the goal cell [29] and vector $\mathbf{n}_a(\mathbf{q}_k, \mathcal{N}_k)$ is computed from repulsive potential functions to achieve inter-robot collision avoidance.

Suppose that $\mathbf{q}_k$ is in cell c. Let $\hat{\mathbf{n}}_e^c$ be the unit vector pointing out of c orthogonal to its exit facet. Let $\hat{\mathbf{n}}_{f_1}^c$, $\hat{\mathbf{n}}_{f_2}^c$ be unit vectors pointing into c orthogonal to each facet adjacent to the exit facet, and define d_{k1}, d_{k2} as the distance from robot k to each of these facets. Also define $\eta, \upsilon, \kappa > 0$. Then

$$\mathbf{n}_g(\mathbf{q}_k) = \eta\ \hat{\mathbf{n}}_e^c + \upsilon\ (1/d_{k1}^{\kappa}\ \hat{\mathbf{n}}_{f_1}^c + 1/d_{k2}^{\kappa}\ \hat{\mathbf{n}}_{f_2}^c)\ .$$

In the last cell in the sequence, this vector is replaced with one pointing from $\mathbf{q}_k$ to the perimeter entrance point.

Let $q_{kl} = ||\mathbf{q}_{kl}|| = ||\mathbf{q}_k - \mathbf{q}_l||$ and $\xi > 0$. Define a sum of vectors that point away from each neighboring robot,

$$\mathbf{n}_n(\mathbf{q}_k, \mathcal{N}_k) = \sum_{l \in \mathcal{N}_k} -\frac{1}{\xi^2 q_{kl}^2}\left(2\ \ln\ (\xi q_{kl}) - \frac{1}{\xi q_{kl}}\right)\mathbf{q}_{kl}\ .$$

This is derived from the example potential function given in [30], with the added parameter ξ that, when lowered, increases the range of repulsion between robots. Finally,

$$\mathbf{n}_a(\mathbf{q}_k, \mathcal{N}_k) = \mathbf{n}_n(\mathbf{q}_k, \mathcal{N}_k)||\mathbf{n}_g(\mathbf{q}_k)||/||\mathbf{n}_n(\mathbf{q}_k, \mathcal{N}_k)||\ .$$

We set the sensing radius ρ to $46\ m$, which is within the capabilities of some laser rangefinders. The navigation speed v_n was set to $1.3\ m/s$, which is attainable by some mobile robots that are particularly suited to surveillance tasks, such as PatrolBot® and Seekur®. The perimeter surveillance speed v_p was set to be 4.5 times slower.

In the optimization problems, the total equilibrium flux capacity c_{tot} for all possible edges (graph Fig. 5b) was set to 0.175 robots/s and distributed among the edges in proportion to the cumulative distance between the centroids of their associated cells.

REFERENCES

[1] B. Gerkey and M. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *Int'l. J. of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[2] L. Vig and J. Adams, "Coalition formation: From software agents to robots," *J. Intellig. and Robotic Syst.*, vol. 50, no. 1, pp. 85–118, 2007.

[3] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artificial Intellig.*, vol. 101, no. 1-2, pp. 165–200, 1998.

[4] L. Vig and J. Adams, "Multi-robot coalition formation," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 637–649, 2006.

[5] M. B. Dias, R. M. Zlot, N. Kalra, and A. T. Stentz, "Market-based multirobot coordination: a survey and analysis," *Proc. of the IEEE*, vol. 94, no. 7, pp. 1257 – 1270, 2006.

[6] M. J. B. Krieger, J.-B. Billeter, and L. Keller, "Ant–like task allocation and recruitment in cooperative robots," *Nature*, vol. 406, pp. 992–995, 2000.

[7] T. H. Labella, M. Dorigo, and J.-L. Deneubourg, "Division of labor in a group of robots inspired by ants' foraging behavior," *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 1, pp. 4–25, 2006.

[8] W. Agassounon and A. Martinoli, "Efficiency and Robustness of Threshold-Based Distributed Allocation Algorithms in Multi-Agent Systems," in *Proc. First Int'l. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2002, pp. 1090–1097.

[9] N. Franks, S. C. Pratt, N. F. Britton, E. B. Mallon, and D. T. Sumpter, "Information flow, opinion-polling and collective intelligence in house-hunting social insects," *Phil. Trans.: Biological Sciences*, vol. 357, pp. 1567–1584, 2002.

[10] O. Shehory, S. Kraus, and O. Yadgar, "Emergent cooperative goal-satisfaction in large-scale automated-agent systems," *Artificial Intellig.*, vol. 110, no. 1, pp. 1–55, 1999.

[11] W. Agassounon, A. Martinoli, and K. Easton, "Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes," *Auton. Robots*, vol. 17, no. 2-3, pp. 163–192, 2004.

[12] K. Lerman, C. V. Jones, A. Galstyan, and M. J. Mataric, "Analysis of dynamic task allocation in multi-robot systems," *Int'l. J. of Robotics Research*, vol. 25, no. 4, pp. 225–242, 2006.

[13] S. Berman, Á. Halász, V. Kumar, and S. Pratt, "Algorithms for the analysis and synthesis of a bio-inspired swarm robotic system." in *Swarm Robotics*, ser. LNCS, vol. 4433. Springer, 2006, pp. 56–70.

[14] ——, "Bio-inspired group behaviors for the deployment of a swarm of robots to multiple destinations," in *Proc. Int'l. Conf. on Robotics and Automation (ICRA)*, 2007, pp. 2318–2323.

[15] Á. Halász, M. A. Hsieh, S. Berman, and V. Kumar, "Dynamic redistribution of a swarm of robots among multiple sites," in *Proc. Int'l. Conf. on Intelligent Robots and Systems (IROS)*, 2007, pp. 2320–2325.

[16] M. A. Hsieh, Á. Halász, S. Berman, and V. Kumar, "Biologically inspired redistribution of a swarm of robots among multiple sites," *Swarm Intelligence*, vol. 2, no. 2, pp. 121–141, 2008.

[17] S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar, "Navigation-based optimization of stochastic strategies for allocating a robot swarm among multiple sites," in *Proc. Conf. on Decision and Control (CDC)*, 2008.

[18] D. Gillespie, "Stochastic simulation of chemical kinetics," *Annu. Rev. Phys. Chem.*, vol. 58, pp. 35–55, 2007.

[19] H. Othmer, *Analysis of Complex Reaction Networks, Lecture Notes*. University of Minnesota, 2003.

[20] J. Sun, S. Boyd, L. Xiao, and P. Diaconis, "The fastest mixing Markov process on a graph and a connection to the maximum variance unfolding problem," *SIAM Review*, vol. 48, no. 4, pp. 681–699, 2006.

[21] R. Russell and T. Urban, "Vehicle routing with soft time windows and Erlang travel times," *J. Operational Research Society*, vol. 59, no. 9, pp. 1220–1228, 2008.
[22] B. Harris, *Theory of Probability*. Reading, MA: Addison-Wesley, 1966.
[23] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.
[24] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge Univ. Press, 1985.
[25] D. P. Landau and K. Binder, *A guide to Monte-Carlo simulations in statistical physics*. Cambridge Univ. Press, 2000.
[26] A. Lewis and M. Overton, "Eigenvalue optimization," *Acta Numerica*, vol. 5, pp. 149–190, 1996.
[27] C. W. Wu, "On Rayleigh-Ritz ratios of a generalized Laplacian matrix of directed graphs," *Linear Algebra and its Applications*, vol. 402, pp. 207–227, 2005.
[28] D. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *J. Comput. Phys.*, vol. 22, pp. 403–434, 1976.
[29] D. C. Conner, A. Rizzi, and H. Choset, "Composition of local potential functions for global robot control and navigation," in *Proc. Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2003, pp. 3546–3551.
[30] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," in *IEEE Transactions on Automatic Control*, vol. 52, no. 5, 2007, pp. 863–868.

Spring Berman (M'07) received the B.S.E. degree in mechanical and aerospace engineering and a certificate in robotics and intelligent systems from Princeton University, Princeton, NJ, in 2005. She received the M.S.E. degree in mechanical engineering and applied mechanics in 2008 from the University of Pennsylvania, Philadelphia, PA, where she is currently working toward the Ph.D. degree in this field.

Her research interests include swarm robotics and bio-inspired control of multi-robot systems.

Ádám Halász (M'06) received the Diploma de Licenta (B.A.) degree in physics from the University of Bucharest, Romania, in 1989. He received the M.A. degree in physics and the Ph.D. degree in nuclear physics from Stony Brook University, Stony Brook, NY in 1995 and 1998, respectively.

He is currently an Assistant Professor in the Department of Mathematics at West Virginia University, Morgantown, WV. He was a Postdoctoral Fellow in the Department of Physics and Astronomy at the University of Pennsylvania from 1998-2003. From 2003-2008, he was a Research Scientist in the GRASP Laboratory at the University of Pennsylvania, where he held an NIH/NLM postdoctoral bioinformatics fellowship from 2004-2006. His research interests include biomathematics, molecular systems biology, mesoscopic phenomena in cells, and swarm robotics.

M. Ani Hsieh (M'96) received the B.S. degree in engineering and the B.A. degree in economics from Swarthmore College, Swarthmore, PA, in 1999 and the Ph.D. degree in mechanical engineering and applied mechanics from the University of Pennsylvania, Philadelphia, PA, in 2007.

She is currently an Assistant Professor in the Mechanical Engineering and Mechanics Department at Drexel University, Philadelphia, PA. Her research interests include bio-inspired control strategies for robot teams, control and coordination of heterogeneous teams, autonomous ground vehicles, and networked robots.

Vijay Kumar (M'87–SM'02–F'05) received the M.S. and Ph.D. degrees in mechanical engineering from The Ohio State University, Columbus, OH, in 1985 and 1987, respectively.

He has been on the Faculty in the Department of Mechanical Engineering and Applied Mechanics with a secondary appointment in the Department of Computer and Information Science at the University of Pennsylvania, Philadelphia, PA, since 1987. He is the UPS Foundation Professor and the Associate Dean for Academic Affairs in the School of Engineering and Applied Science. He served as the Deputy Dean of the School of Engineering and Applied Science from 2000-2004. He directed the GRASP Laboratory, a multidisciplinary robotics and perception laboratory, from 1998-2004. He was the Chairman of the Department of Mechanical Engineering and Applied Mechanics from 2005-2008. His research interests lie in the area of robotics and networked multi-agent systems.

Dr. Kumar has served on the editorial boards of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, *Journal of the Franklin Institute*, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, ASME *Journal of Mechanical Design*, and ASME *Journal of Mechanisms and Robotics*. He is the recipient of the 1991 National Science Foundation Presidential Young Investigator award, the Lindback Award for Distinguished Teaching, the 1997 Freudenstein Award for significant accomplishments in mechanisms and robotics, and the 2004 IEEE International Conference on Robotics and Automation Kawamori Best Paper Award. He is also a Distinguished Lecturer in the IEEE Robotics and Automation Society and an elected member of the Robotics and Automation Society Administrative Committee. He is a Fellow of the American Society of Mechanical Engineers.